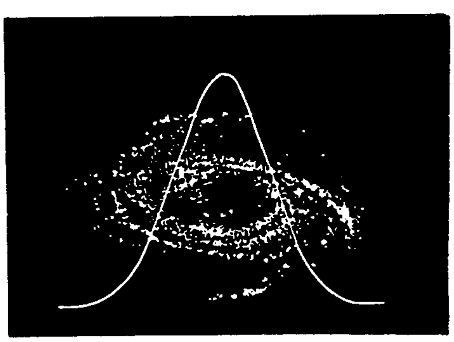
AR0 32850.12-MA

Statistical Software, Siftware and Astronomy

Edward J. Wegman, Daniel B. Carr, R. Duane King,
John J. Miller, Wendy L. Poston,
Jeffrey L. Solka and John Wallin

Technical Report No. 128
May, 1996

Center for Computational Statistics



George Mason University
Fairfax, VA 22030

CENTER FOR COMPUTATIONAL STATISTICS
TECHNICAL REPORT SERIES

TR 111. Winston C. Chow, Fractional Process Modeling, November, 1994.

TR 112. Mark C. Sullivan, ***Computationally Efficient Statistical Signal Processing Using Nonlinear Operators*** (Ph.D. Dissertation), December, 1994.

TR 113. Irwin Greenberg, Some Simple Approximation Methods in Level Crossing Problems, December, 1994.

TR 114. Jeffrey L. Solka, Wendy L. Poston and Edward J. Wegman, A New Visualization Technique to Study the Time Evolution of Finite and Adaptive Mixture Estimators, December, 1994. published ***Journal of Computational and Graphical Statistics***, 4(3), 180-198, 1995.

TR 115. D. B. Carr and A. R. Olsen, Representing Cumulative Distributions with Parallel Coordinate Plots, August, 1995

TR 116. Jeffrey L. Solka, ***Matching Model Information Content to Data Information*** (Ph.D. Dissertation), August, 1995.

TR 117. Wendy L. Poston, ***Optimal Subset Selection Methods***, (Ph.D. Dissertation), August, 1995.

TR 118. Clifton D. Sutton, Sphere Packing, August, 1995.

TR 119. Wendy L. Poston, Edward J. Wegman, and Jeffrey L. Solka, A Parallel Algorithm for Subset Selection, August, 1995.

TR 120. Barnabas Takacs, Harry Wechsler, and Edward J. Wegman, A Model of Active Perception and its Implementation on the Intel Paragon XP/S, August, 1995.

TR 121. Shan-chuan Li, Walter Dyar, and Mary-Ellen Verona, GRASS Database Explored and Applied to Biodiversity Query with Splus, August, 1995, to appear ***Computing Science and Statistics***, 27, 1995.

TR 122. Kathleen Golitko Perez-Lopez, ***Management of Scientific Image Databases Using Wavelets*** (Ph.D. Dissertation), August, 1995.

TR 123. Edward J. Wegman, Jeffrey L. Solka and Wendy L. Poston, Immersive Methods for Mine Warfare, April, 1996.

TR 124. Edward J. Wegman and Qiang Luo, High Dimensional Clustering using Parallel Coordinates and the Grand Tour, April, 1996.

TR 125. Kletus A. Lawler, ***Linear and Nonlinear Regression Estimates for a Cobb-Douglas Model***, (M.S. Thesis), April, 1996.

TR 126. Ehsan S. Soofi, Information Theoretic Regression Methods, April, 1996.

TR 127. Celesta Ball and Edward J. Wegman, Geometric Modeling of Vehocle Paths and Confidence Regions, May, 1996

TR 128. Edward J. Wegman, Daniel B. Carr, R. Duane King, John J. Miller, Wendy L. Poston, Jeffrey L. Solka, and John Wallin, Statistical Software, Siftware, and Astronomy, May, 1996

Statistical Software, Siftware and Astronomy[1]

Edward J. Wegman, Daniel B. Carr, R. Duane King, John J. Miller,
Wendy L. Poston, Jeffrey L. Solka, and John Wallin

George Mason University
Fairfax, VA 22030-4444

Abstract

This paper discusses statistical, data analytic and related software that is useful in the realm of astronomy and spaces sciences. The paper does not seek to be comprehensive, but rather to present a cross section of software used by practicing statisticians. The general layout is first to discuss commercially available software, then academic research software and finally some possible future directions in the evolution of data-oriented software. We specifically exclude commercial database software from the discussion, although it is relevant. The paper focuses on providing internet (world wide web) pointers for a variety of the software discussed.

1. Introduction

It seems somewhat presumptuous for a group of statisticians (and one astronomer) to tell a group of astronomers what manner of statistical software they need. The alternative is an attempt at an encyclopedic cataloguing of existing statistical software, an effort that would seem to have little value added. Fortunately, there are a few guides to the type of statistical methods perceived by astronomers as being required. An early work by Trumpler and Weaver (1953) is based on lectures given in 1935 and focuses on the application of then emerging statistical theory to astronomy. While traditional statistical theory is exposited, applications even then focus on statistical techniques for spectral distributions and spatial distribution of stars. These presage elements of time series analysis and spatial statistics, the latter being particularly a topic of considerable interest

---

[1]The work of the principal author, Dr. Wegman, was supported by the Army Research Office under contract DAAH04-94-G-0267. The associate authors are listed in alphabetical order of last names. Drs. Wegman, Carr and Miller are members of the Department of Applied and Engineering Statistics at George Mason University and are also affiliated with the Center for Computational Statistics at GMU. Dr. Wallin and Mr. King have primary affiliations with the Institute for Computational Sciences and Informatics at GMU. Drs. Poston and Solka have primary affiliation with the Naval Surface Warfare Center in Dahlgren, VA. All of the authors have an affiliation with the Institute for Computational Sciences and Informatics. The work of Dr. Carr was support be the Evironmental Protection Agency. The work of Drs. Poston and Solka was supported by the Office of Naval Research under the ILIR program.

among statisticians. Interest in spatial point processes is also reflected in the much more recent collaborative work by Babu and Feigelson (1996). Slightly earlier work by Rolfe (1983) and by Murtagh and Heck (1988) carried strong elements of analysis based on large databases reflecting current statistical interest in massive data sets. Jaschek and Murtagh (1989) introduce concerns of data analysis and fitting, and of small sample issues suggesting bootstrapping and jackknifing techniques. Perhaps the most definitive articulation of the role of statistics in astronomy is Feigelson and Babu (1992). They identify work cluster analysis, truncation and censoring, Bayesian methods and image analysis, time series analysis, and multivariate methods. We would perhaps add to the themes articulated above graphical exploratory analysis and visualization. Our attempt to describe statistical and related software will be built around these statistical and related computing themes:

Spatial statistics and spatial point process
Time series analysis, spectral distributions
Massive data sets, databases
Clustering methods, pattern analysis
Truncation and censoring
Image analysis, particularly Bayesian methods
Multivariate methods
Visual exploratory analysis.

We divide our discussion into the deliberately provocative eras: Past, Present and Future. In our discussion, we really take **Past** to be synonymous with **commercially available software**, **Present** to be synonymous with **academic and research software** that is not commercially supported, and **Future** to be **not yet existing software** but software that seems likely to be required based on a little speculation on the likely nature of future requirements. Of course, commercially available software is legacy software for which there is a major investment in both people and existing databases. The companies that release commercial software have a very big infrastructure to support that software and do an excellent job in keeping up with the latest developments. However, because of their required adherence to data structures and styles of computing along which their analysis systems were originally developed, they lack much of the agility that academic and research code can exploit. Academic code, on the other hand is developed with much less discipline and is traditionally unsupported or supported comparatively poorly. Our speculation on future code may, of course, ultimately prove to foolish. Yet it does perhaps point the way to what we should at least expect.

As turbulent as computing has been since the introduction of microprocessor-based personal computers and workstations, we are on the threshold of an even more uncertain and exciting era. Several phenomena are worth noting:

1) Within the last 18 months, the face of supercomputing has changed dramatically. Cray Computers went bankrupt, Cray Research was sold to Silicon Graphics, Convex Computers became a wholly owned subsidiary of Hewlett-Packard, and Intel

announced that they would cease production of their Paragon, the current holder of world speed record. Most of the other start-ups producing supercomputer-class machines are in bankruptcy or have ceased doing business altogether. Currently, a 90 megahertz Pentium PC has essentially the same computing power as the $12,000,000 Cray 1 supercomputer did in 1975. Wegman (1995) compares the performance of algorithms of several levels of computational complexity over a number of sizes of data sets and concludes that for most problems, personal and workstation computing is adequate. Moreover, feasibility of many techniques is limited by data transfer rates and the limits of visualization, rather than by computational horsepower.

2) The ubiquity of the world wide web is a phenomenon none can escape. (Wegman has a collection of free introductory disks and CDs from America Online, CompuServe, and other vendors of internet services that measures nearly two feet in length.) Having a web page is now a mode of business as crucial as having a fax machine was three years ago. A business now seems to be at a serious disadvantage if it doesn't have a web page.

3) High performance computing which, at one time, was essentially synonymous with vector processor supercomputers solving partial differential equations has been broadened to include not only computationally-intensive applications, but also data-intensive and information-intensive applications. The high speed network becomes even more crucial in this context. Several gigabit testbeds are currently being demonstrated and ATM/Sonet fiber optic based networks will become the standard. Current ethernet is 10 megabits per second, but near-term future networks will be OC-3 to OC-48 (155 megabits per second to 2.48 gigabits per second). The notion of a *hollow computer*, a personal computer or workstation for which most of the computations are done transparently by other machines out on the network, will become a reality.

The software discussed in this paper is selected for discussion based on highly personalized experience. It reflects a cumulative experience of a number of people who really do statistical and scientific computing on a daily basis, hence our multiple authorship. We certainly do not guarantee that we are inclusive in describing all possible packages. Earlier works by Francis (1981) and by Hayes (1982) survey respectively 60 and 213 statistical packages. A summary of the latter work appeared as Wegman and Hayes (1988), but all of these are hopelessly outdated now and are only of historical interest. Rather than an all-inclusive survey, we choose to describe a few of the packages we find useful in day-to-day computing.

Section 2 describes some of the statistical, mathematical and visualization packages that are commercial available, that is, our *Past* section. Section 3 describes some of the academic and research packages and is our *Present* section while Section 4 describes some potential *Future* software/siftware. In all of our discussions, we give useful URLs, where available, for each of the software packages. Of course, these will be updated, and hence contain more recent information about the software under consideration. However, their long-term stability is less certain. Nonetheless they will be

starting points extending the useful life of this paper. We conclude this paper with general remarks in Section 5.

2. Past: Commercially Available Software

We believe there are three general classes of software available using several different user interfaces. Statistical software begins to blend in one direction with relational database software such as Oracle or Sybase (software we do not discuss here) and with mathematical software such as MATLAB in the other direction. Mathematical software exhibits not only statistical capabilities flowing from code for matrix manipulation, but also optimization and symbolic manipulation useful for statistical purposes. Finally visualization software overlaps to some extent with software intended for exploratory data analysis. The user interfaces common range from command line to graphical user interfaces (GUI) to hybrid drag and drop system interfaces. We cast our net fairly widely in describing commercial software because of the general boundary crossing capabilities of the software systems.

The SAS® System for Statistical Analysis

SAS began as a statistical analysis system in the late 1960's growing out of a project in the Department of Experimental Statistics at North Carolina State University. The SAS Institute was founded in 1976. Since that time, the SAS System has expanded to become an ever evolving system for complete data management and analysis. Among the many products making up the SAS System are products for: management of large data bases; statistical analysis of time series; statistical analysis of most classical statistical problems, including multivariate analysis, linear models (as well as generalized linear models), and clustering; data visualization and plotting. A geographic information system is one of the products available in the system. The SAS System is available on PC and UNIX based platforms, as well as on mainframe computers.

One may use the SAS system to conduct simulation studies with random number generators for many different distributions. Managing very large data bases is easy. One may subset, merge, and rearrange databases with comparative ease. Using modern database techniques with queries is also very easy. Data transformation is also accomplished easily. User written functions can be integrated into the system. A product of the system useful for these purposes is SAS/BASE. Programs are written in a language which resembles PL/1 or C. This level of programming is not required unless the user wishes to perform sophisticated transformations. Many applications can be accomplished using simple point and click operations. For users with a need to write an applications program using a matrix language, the product SAS/IML provides the ability to program using matrices as objects.

Data may be imported into and exported from SAS using the SAS/ACCESS product. In PC SAS, data may be imported from most commercial spread sheet or database software. The SAS/STAT product allows the user to analyze many types of data. Linear models (regression, analysis of variance, analysis of covariance), generalized linear models (including logistic regression and Poisson regression), multivariate methods (MANOVA, canonical correlation, discriminant analysis, factor analysis, clustering), categorical data analysis (including log-linear models), and all standard techniques for descriptive and confirmatory statistical analysis. The statistical analyses may be interfaced with the graphical products to produce relevant plots such as q-q plots, residual plots, and other relevant graphical descriptions of the data. The graphical product is SAS/GRAPH.

The SAS/ETS product allows the user to accomplish sophisticated analyses of time series data. Multiple time series and linear systems of time series can be analyzed. Again, relevant plots can be made via interaction with SAS/GRAPH. ARIMA models and state space models are covered. Spectral analysis may also be done. The SAS/INSIGHT product accomplishes exploratory data analysis using graphical display of results. Many statistical analyses available in other products are present in this module linked to visualization displays. Brushing and other dynamic graphical techniques may be used. Animated 3-d plots are available. Box plots, scatterplot matrices, distribution plots, and other visualization tools are present. The SAS/GIS product is a geographic information system built in to the larger SAS system. Spatial data may be stored, linked, analyzed, and displayed using SAS/GIS.

SAS is to a large extent an industry standard statistical software package. We find that demand for students with SAS skills is considerably greater than for students with skills other statistical packages. This may be an artifact of our location on the East coast as well as SAS's East coast location. However, even in a European setting we see considerable demand for SAS. Some useful URL's are **http://www.sas.com/** which is the main URL for SAS and also **http://is.rice.edu/~radam/prog.html** which contains some user-developed tips on using SAS. Web search engines also can turn up many, many references to SAS. An AltaVista **(http://altavista.digital.com/)** search on SAS turns up more than 40,000 hits.

Other statistical systems which are of the same general vintage as SAS are MINITAB, BMDP and SPSS. All of these systems began as mainframe systems, but have evolved to smaller scale systems as computing has evolved.

MINITAB

Minitab Inc. was formed more than 20 years ago around its flagship product, MINITAB statistical software. MINITAB Statistical Software provides tools to analyze data across a variety of disciplines, and is targeted for

users at every level: scientists, business and industrial users, faculty, and students. Originally formed to create software that helped faculty teach basic statistics, the company has broadened the scope of its products to include quality control, designed experiments, chemometrics, and an extensive array of general statistics. Minitab Inc.'s flagship product is MINITAB Statistical Software. MINITAB is available on the most widely-used computer platforms, including Windows, DOS, Macintosh, OpenVMS, and Unix. The Student Edition of MINITAB, is distributed internationally by Addison-Wesley Publishing Company, and is widely used by college and high school students. MINITAB is quite easy to learn and use, right out of the box. Pull-down menus and dialog boxes give you easy prompts every step of the way. There's no lengthy learning process and little need for unwieldy manuals. Most usage is by double-clicking on a program icon. Data are entered in MINITAB's data window and may be imported directly from a variety of file formats, including Lotus, Excel, Symphony, Quattro Pro, dBase and text (ASCII) files. MINITAB macros allow the creation of custom operations. DO loops, IF THEN ELSE statements and GOTO statements are available. Because Release 10 Xtra is available for both Macintosh and Windows machines, one can transparently transition from the Macintosh version of MINITAB to the Windows version (and vice versa) since output and screens are nearly identical. The URL for MINITAB is **http://www.minitab.com/**. An AltaVista search on MINITAB turns up about 1000 hits.

BMDP

BMDP has its roots as a bio-medical analysis packages from the late 1960s.. Current versions come in several flavors including the BMDP New System Personal Edition, the BMDP Classic for PCs - Release 7, and the BMDP New System Professional Edition. BMDP New System has an easy-to-use interface that makes data analysis possible with simple point and click and fill-in-the-blank interactions. Pop-up windows and dialog boxes will then prompt the user until the analysis is complete. The Professional Edition combines the full suite of BMDP Classic for PCs Release 7 statistics with the powerful data management and front-end data exploration features of the BMDP New System Personal Edition. Featuring a comprehensive library of over forty statistical routines, BMDP Classic has set the standard for high-end statistical analysis software. Each statistical routine has been thoroughly time-tested and is based on the most advanced algorithms available. Visualization tools include a customizable plotting utility, linkage between all personal edition plots, datasheet, and statistical output, and a number of standard plots and charts. The BMDP New System Personal Edition includes descriptive statistics, t-tests, nonparametric tests, one-way and two-way ANOVA, frequency tables, and simple and multiple regression with comprehensive diagnostic statistics. BMDP New System Professional Edition adds multi-way description of groups, log-linear modeling, correspondence analysis, regression

(including stepwise, regression on principal components, and ridge regression), non-linear regression, analysis of variance, multivariate analysis (including factor analysis, cluster analysis of cases, variables, and blocks), time series and survival analysis. An AltaVista search of BMDP turns up about 2000 hits. A reference URL for BMDP is **http://www.ppgsoft.com/bmdp00.html**.

SPSS

SPSS is a multinational software company that provides statistical product and service solutions for survey research, marketing and sales analysis, quality improvement, scientific research, government reporting and education. SPSS software products run on most models of all major computers. In the late 1960s, SPSS Chairman of the Board Norman H. Nie, C. Hadlai (Tex) Hull and Dale Bent, three Stanford University graduate students, developed the SPSS statistical software system. In 1968, Nie and his colleagues founded SPSS. In 1975, SPSS incorporated and established headquarters in Chicago, where the company remains today. In August 1993, SPSS became publicly traded. In the 1970s and through the mid-1980s, SPSS software operated primarily on large computing systems, which limited access of the product to large academic institutions and corporations. Then in 1984, SPSS introduced SPSS/PC+ for computers running MS-DOS. Statistical analysis could now be done on the desktop. SPSS released products for UNIX systems in 1988 and for Macintosh in 1990. SPSS is intended as a complete tool kit of statistics, graphs and reports. SPSS starts with the SPSS Base which includes most popular statistics, complete graphics, broad data management and reporting capabilities. The SPSS products are a modular system and includes SPSS Professional Statistics, SPSS Advanced Statistics, SPSS Tables, SPSS Trends, SPSS Categories, SPSS CHAID, SPSS LISREL 7, SPSS Developer's Kit, SPSS Exact Tests, Teleform, and MapInfo. An AltaVista search for SPSS turns up about 10,000 hits. A reference URL for SPSS is **http://www.spss.com/**.

MINITAB is used extensively in the educational community. Indeed, we use it for our introductory courses. BMDP and SPSS tend to find users among the communities in which they originated, respectively the biomedical community and the social sciences community. While this may be a value judgment from our perspective, it appears that mainstream applied statisticians tend to use SAS more extensively. Splus on the other hand seems to be a package that is highly regarded among the more research oriented statisticians, particularly those interested in computational statistics.

S-PLUS

While there are many different packages for performing statistical analysis, one that offers some of the greatest flexibility with regard to the implementation of user defined functions and the customization of ones

environment is S-PLUS. S-PLUS can be thought of as a high-level programming language which has been designed for the easy implementation of statistical functions. Besides excellent support for statistical and user defined operations this language offers the user extensive graphics and hardcopy capability.

S-PLUS is a supported extension of the statistical analysis language S. S was originally developed at AT&T by a team of researchers including Richard A. Becker, John M. Chambers, Allan Wilks, William S. Cleveland and Trevor Hastie. The original description of the S language was written by Becker, Chambers, and Wilks (1988). A good introduction to the application of S to statistical analysis problems is contained in Statistical Models in S by Chambers and Hastie. More recent work that focus on the statistical capabilities of the S-PLUS system include Modern Applied Statistics with S-PLUS by Venables and Ripley and Statistical Analysis in S-Plus by StatSci.

S-PLUS is manufactured and supported by the Statistical Sciences Corporation, now a division of MathSoft. The development of S-PLUS has for the most part been supervised by R. Douglas Martin. There have been many other researchers who have worked on the S-PLUS extension. Some of the code has even been contributed by prominent individuals from the academic and industrial communities. For example Rob Tibshirani and Jerome Friedman have contributed some of the code that resides in the package. Much of this code resides at Carnegie Mellon University under their Statlib library system. The reader may obtain an index of the S Statlib software by sending email to **statlib@lib.stat.cmu.edu** with the single line message send index from S.

S-PLUS runs on both PC and UNIX based platforms. In addition the company offers easy links for the user to call S-PLUS from within C/FORTRAN or for the user to call C/FORTRAN compiled functions within the S-PLUS environment. Statistical Sciences has made great efforts to keep the software current with regard to the needs of the statistical community. They have released dedicated modules which are targeted at specific application areas. For example in the Summer of 1995 they released a wavelets module and at the time of the writing of this article they had planned to release a spatial statistics module.

The S-PLUS package provides the user with a plethora of statistical capabilities. These include the ability to generate random data from 20 different distributational types and the ability to perform 12 different types of hypothesis tests including Student's t-test and the Wilcoxon test. It allows one to perform linear, nonlinear, and projection pursuit regression. There are also capabilities for multivariate analysis including graphical methods, cluster analysis and discriminant analysis. In addition there are the standard time series analysis tools for ARIMA models and seasonally adjusted data. Finally we would like

to point out that the graphical display capabilities are well developed as is to be expected since William Cleveland has had a strong role in shaping the format of the S-PLUS graphical functions.

The capabilities inherent in the spatial statistics module may be of particular interest to the astronomical community. These include analysis tools for spatial point patterns, area data, and spatially continuous data. Spatial point patterns are those data sets that consists of indicators of a random process that has occurred at particular locations in space. For example the location where a particular disease has manifest itself. One is often interested on whether these manifestations are clustered in a spatio-temporal sense. Area data would occur if we were to collapse these disease outbreak indicators form the latitude longitude level to the level of counties or states. So the information in this case would be at a courser level of fidelity than that associated with the point process. The object of analysis could still be the spatial relationship among these county level epidemiological counts. In the case of spatially continuous data one has measured a process on a regular or an irregular grid. In this case one is interested in the relationship of the measured variable as the grid location is varied.

One can easily imagine how these data types would manifest themselves within an astronomical framework. Sky catalog images can be viewed as spatial point patterns on one level. Remote sensing planetary information might fall into the realm of spatially continuous data or area data depending on the nature of the collection process. Hence the S-PLUS spatial capabilities may be quite useful for the analysis of these data types. An AltaVista search on S-PLUS turns up about 1000 hits.

The S-PLUS home page can be reached at **http://www.mathsoft.com/.** The URL: **http://www.gcrc.ufl.edu/gopher.documents/sas/sas.vs.splus.html** features a comparison between SAS and S-PLUS.

Other statistically oriented packages enjoying good reputations are SYSTAT, DataDesk, and JMP. SYSTAT originated as a PC-based package developed by Leland Wilkinson. SYSTAT is now owned by SPSS and more information on SYSTAT can be found at URL **http://www.spss.com/**. SYSTAT has about 1000 AltaVista hits. The current version is 6.0 and is a Microsoft Windows oriented product. DataDesk is a Macintosh-based product authored by Paul Velleman from Cornell University. Currently released is version 5.0.1. This is a GUI-based product which contains many innovative graphical data analysis and statistical analysis features. More information about DataDesk can be found at URL: **http://www.lightlink.com/datadesk/**. DataDesk has about 200 AltaVista hits. JMP is another SAS product that is highly visualization oriented. JMP is a stand alone product for PC and Macintosh platforms. It originated as a Macintosh product and resembles DataDesk in some ways. Information on JMP can be found at **http://www.sas.com/**. An AltaVista query on JMP is indeterminate since this

abbreviation appears to have many other meanings. While more could be written about these individual products (and probably should be), we leave the discussion at this stage.

The descriptions of statistical software above cover the most well-established commercially available software packages. Mathematical packages often exhibit some statistical capabilities, especially when engineering or other basic science applications have overlap with statistics. Among the most extensively used mathematical packages is MATLAB. MATLAB has many features that resemble APL, a language popular for statistical computing in the 1970s.

MATLAB

MATLAB is an interactive computing environment that can be used for scientific and statistical data analysis and visualization. It is similar to the data analysis software IDL that may already be familiar to many researchers in astronomy. The basic data object in MATLAB is the matrix. The user can perform numerical analysis, signal processing, image processing and statistics on matrices, thus freeing the user from programming considerations inherent in other programming languages such as C and FORTRAN. There are versions of MATLAB for Unix platforms, PC's running Microsoft Windows and Macintosh. Because the functions are platform independent, provides the user with maximum reusability of their work.

MATLAB comes with many functions for basic data analysis and graphics. Most of these are written as M-file functions, which are basically text files that the user can read and adapt for other uses. The user also has the ability to create their own M-file functions and script files, thus making MATLAB a programming language. The recent addition of the MATLAB C-Compiler and C-Math Library allow the user to write executable code from their MATLAB library of functions, yielding faster execution times and stand-alone applications.

For researchers who need more specific functionality, MATLAB offers several modules or toolboxes. These typically focus on areas that might not be of interest to the general scientific community. Basically, the toolboxes are a collection of M-file functions that implement algorithms and functions common to an area of interest. Some of the toolboxes that would be useful in astronomy are Statistics, Signal Processing, Image Processing and Symbolics. The Statistics Toolbox performs basic hypothesis tests, regression, and statistical visualization. The Signal and Image Processing Toolboxes include functions for signal display, filtering and analysis. The Symbolics Toolbox contains the Maple Kernel and comes in a basic collection of functions or an extended version which includes the Maple programming features. There are also several third-party packages that are available, including a package called Wavbox that implements wavelet analysis algorithms.

One of the most useful capabilities of MATLAB is the tools available for visualizing data. There are many 2-D and 3-D plotting functions such as surface and mesh plots, contour plots, histograms, and image plots. These are provided in high-level functions where the plotting details are hidden surface and mesh plots, contour plots, histograms, and image plots. These are provided in high-level functions where the plotting details are hidden from the user. However, for those who require total control over their plots, MATLAB provides Handle Graphics. These are a set of graphics objects and their corresponding properties. These properties can be changed as desired. MATLAB also provides graphical objects that allow the user to create Graphical User Interfaces (GUI's). These objects include sliders. buttons, and menus. These are extremely useful tools for people who want to package their algorithms in something that is powerful, reusable and easy to use.

There is a considerable amount of contributed MATLAB code available on the internet. One notably useful source for astronomers is the MATLAB Astronomy Library at the Astronomy Department of the University of Western Ontario. This library has M-file functions that have been developed by the department for analyzing astronomical data. This site can be accessed at **http://phobos.astro.uwo.ca/~etittley/matlab/matlab-astrolib.html**. Another source of code is available via the home page for MATLAB at **http://www.mathworks.com**. MATLAB has more than 10,000 hits in an AltaVista search.

MATLAB is used extensively in our University and is a particular favorite of engineers, physicists and chemists. Other mathematical software worth noting is *Mathematica* and MAPLE, both of which have powerful symbolic processing capabilities. *Mathematica* also has numerical and graphical features, but is comparatively complex to learn. Information on *Mathematica* is available at URL **http://www.wolfram.com/** while additional information on MAPLE is available at **http://www.maplesoft.com/**. An AltaVista search on *Mathematica* turns up some 40,000 hits while a search on MAPLE turns up 7,000 hits many of which refer to trees. Another useful mathematical package is MATHCAD, a package which combines numerical, symbolic, and graphical features. MathSoft, Inc., producers of MATHCAD, have recently acquired S-Plus, so that information on MATHCAD is also available at **http://www.mathsoft.com/.** MATHCAD turns up about 3,000 hits in an AltaVista search. A longtime standard package that spans both statistical and mathematical techniques is the IMSL scientific subroutine library. Most scientists are probably familiar with IMSL. The IMSL corporation merged several years ago with the producers of PVWave and is now known as Visual Numerics, Inc. More information on both of these products can be obtained at **http://www.vni.com/**. A good source of expertise and helpful hints for IMSL users is available at **http://www-c8.lanl.gov/dist_comp2/MATH/Imsl/imsl_keyword.html**.

Visualization tools are becoming more powerful and hence more useful for the statistician and data analyst. S-Plus, DataDesk, JMP and more recently MATLAB are incorporating advanced visualization tools. We have alluded to PVWave above as a visualization packages. Other advanced visualization tools include AVS and IDL. As indicated above, more information on PVWave is available at **http://www.vni.com/.** Information on AVS is available at **http://www.avs.com/.** AVS has an extensive users group and a library of software applications and other information for AVS is available at **http://testavs.ncsc.org/**. Information on IDL is available at **http://www.rsinc.com/**. (The latter organization also produces the commercial version of the visible human project, a massive database on high resolution, three-dimensional human anatomy.) URL, **http://axp2.ast.man.ac.uk:8000/~dsb/visual/sg8.htx/node16.html**, is a general resource for visualization packages.

3. Present: Academic and Research Software

To state what is obvious, the down-side of academic and research software is that it tends to be less comprehensive and less reliable than commercial-grade, supported software. The upside is that it tends to be more innovative and daring in concept. Of course, most of the commercial software discussed in Section 2 has roots in academic, research software. Because academic, research software is generally not as widely distributed, our discussion of it will be more limited in scope and personalized in perception. We discuss four academic, research packages: 1) XGobi, 2) Xlisp-Stat, 3) ExplorN, and 4) Manet.

XGobi

XGobi is an inactive high-dimensional visualization package. This X-Window-based system (X-Windows is a trademark of MIT) implements many useful data exploration concepts developed or promoted by the statistical graphics community during the two decades. The concepts include focusing via rescaling, conditioning and sectioning; point linking across and rearrangement of multiple views, direct interactive manipulation including stretching, panning, zooming, rotation, point identification, and brushing; projection and section tours, data transformations, and algorithms for finding optimized views. The thoughtful integration of this methodology in XGobi makes it an outstanding package for revealing structure in multivariate data.

XGobi, is part of the continuing effort of the statistics community to make powerful methods accessible to the scientific community. The history behind XGobi can in part be found in Cleveland and McGill (1988). Papers by Fisherkeller, Friedman and Tukey (1974), Friedman and Tukey (1974), Donoho, Huber and Thoma (1981), McDonald (1982), Becker and Cleveland (1984), and Asimov 1985) cover many of the developments from decade ago. The methods opened the door to much deeper investigation of high-dimensional data but often were slow in reaching the scientific community.

XGobi is result researchers' efforts (see Swayne, Cook and Buja, 1992) to make the powerful methodology available to the public. XGobi is free over the network. The documentation is now sold for a modest price.

As suggested above XGobi provides univariate dot plots, XY plots, rotation, grand tour, projection pursuit guided tour with many choice for the projection pursuit index, linked plot brushing and more. While XGobi is both fun and easy to use, knowledge about how seek and interpret structure is important. A body of research provides the foundation for understanding multivariate structure using XGobi. Furnas and Buja (1994) provide an enlightening discussion on the discovery of the dimensionality of objects embedded in high dimensional space via low dimensional views. For example straight lines in p-dimensions project as straight lines in 2-D. Solids in 3-D will saturate the plot when projected into 2-D. With appropriate knowledge and procedures one can identify the dimensionality of objects (through 6-dimensions) using scatterplots. Buja, Cook, and Swayne (1996) continue in their efforts to guide users so that the simple tools become powerful tools of understanding. For example the linked brushing of dendrograms helps to provide understanding of clustering algorithm results and to suggest alternative clusters.

The dimensionality that XGobi can practically handle is limited. The words, *high-dimensional*, here means something like 20 variables and not 500 variables. For very high-dimensional problems, dimension reduction methods are required before XGobi is useful. As always, scientifically insightful transformations and dimension reductions can be crucial. The structure one looks for depends on the type of problem. Some times the problem is related to data density and the interest is in clusters, outliers, and holes. Other times the problem is one of linear or nonlinear regression, and graphical forms of sliced inversion regression can shed insight. That one can actually understand fully 10 dimensional data is unlikely. However, a surprisingly common situation is that low-dimensional structure is embedded in high-dimensional data. For example satellite spectral intensity bands are one source of multivariate data. For views of the ground, each pixel's multivariate values are often mixtures because the pixel covers a mixture of vegetation types, bare-land types and water of varying clarity. Mixture of two distinctive types generate linear structure. Mixtures of three distinctive types generate a triangular structure and so on. When high dimensional data is composite of fairly simple low-dimensional structures, humans can understand a lot. The key is to have the right set of clustering, projection and sectioning tools that help find and isolate understandable constituent elements.

The efforts to extend the domain of application of XGobi continue. Versions of XGobi can communicate with ARC/VIEW TM (and ARC/INFO TM) via remote procedure calls (Symanzik, Majure, and Cook 1995). One can

sample a satellite image or a coverage in ARC/VIEW and use XGobi to search for structure in attribute space. Brushed points in XGobi change color in the ARC/VIEW map and vice versa. XGobi provides insightful reexpression of data coming from ARC/VIEW. For example, spatial statistics methodology often uses the variogram to assess spatial correlation. XGobi will display a variogram. The process involves calculating values from points pairs. If one also calculates and sin and cos for the directions defined by point pairs and plots those two coordinates the result is a circle. Brushing the circle reveals the directional variograms embedded in the linked variogram display. If one is interesting in comparing subsets created by brushing, one can even evaluate the subsets by graphically comparing calculated multivariate CDFs (see Majure et al 1995). Both thought and cleverness have gone into the development and extensions of XGobi capabilities. A reference URL for XGobi is **http://lib.stat.cmu.edu/general/XGobi/.index.html**. An AltaVista search on XGobi returns more than 550 hits.

Xlisp-Stat

Xlisp-Stat is an object-oriented environment for statistical computing and dynamic graphics. Written by Professor Luke Tierney, School of Statistics, University of Minnesota, Xlisp-Stat was motivated by the "S" system, with the basic principal that an extendible system is necessary for conducting research on new computationally based statistical methods. Xlisp-Stat provides a set of high-level tools to develop new dynamic graphics techniques. Although motivated by S, Xlisp-Stat is based on Lisp, a well-established, complete and flexible programming language. Like S, Xlisp-Stat is an interpreted language, which is much more suited towards exploration than a compiled language such as C/C++ or Pascal. These require lengthy recompilations to fix bugs or test simple new ideas. When greater speed is required, a byte-code compiler can be run from inside Xlisp-Stat. This compiler can give increase the speed of execution by an order of magnitude. The defining reference book on Xlisp-Stat is Tierney (1990). Since Xlisp-Stat is founded on Lisp, almost any book on general Lisp programming can help you get started.

Xlisp-Stat is available for Unix/X Windows, Macintosh/MacOS and for IBM PC compatibles running Windows 3.1. Xlisp-Stat is freeware, available by anonymous ftp to **ftp.stat.umn.edu**. The Macintosh version of Xlisp-Stat requires System 6 or later, 3 MB of free hard disk space and at least 5 MB of RAM. The actual amount of RAM required depends on how one plans to use Xlisp-Stat and on the Macintosh's display type. Xlisp-Stat will run on any machine with that bare memory requirement even a Mac Plus, but machines with at least a 68020 are recommended to make Xlisp-Stat useful (although a 68020 will still not yield a system capable of running large simulations). The Windows 3.1 version of Xlisp-Stat requires at least 5 MB of RAM, 3 MB free of hard disk space and Windows 3.1 or later. There are two versions available:

a 16 bit version and 32 bit version; the 16 bit version runs under OS/2. The 32 bit version requires version 1.15 of Microsoft's Win32s (or later), Windows95 or WindowsNT. For Unix workstations, the R-code project **(http://www.stat.umn.edu/~bjm/rcode/index.html)** has information on retrieving and compiling Xlisp-Stat for Unix. You need a workstation running X11R4 or later.

Xlisp-Stat is based on Xlisp by David Betz, which is a variant of Lisp. In late 1994 Xlisp-Stat has made great strides towards becoming Common Lisp-compliant, so most books on Lisp can help you get started in Xlisp-Stat. The defining reference on Common Lisp is Guy Steele's "Common Lisp." This is not an introduction to Lisp programming, but is the complete definition of Lisp. An on-line version of Steele's "Common Lisp" is available **(http://www.cs.cmu.edu/Web/Groups/AI/html/cltl/clm/clm.html)**. Another general resource is the Association of Lisp Users home page **(http://www.cs.rochester.edu/users/staff/miller/alu.html)** they include a list of books on Lisp that might be worth checking before purchasing one.

There are several sources of further information. UCLA's Xlisp-Stat Archive **(http://www.stat.ucla.edu/develop/lisp/)** is run by Jan Deleeuw of the UCLA Department of Statistics. This is the largest public collection of Xlisp-Stat code. Penn State University's Lisp-Stat page includes descriptions of projects around the world using Xlisp-Stat **(http://euler.bd.psu.edu/lispstat/lispstat.html)**. Another forum for questions about Xlisp-Stat is the `lisp-stat-news` mailing list available by sending email to **luke@stat.umn.edu**. An AltaVista search on Xlisp-Stat turns up 300 hits.

ExplorN

ExplorN is a statistical graphics and visualization package designed for the exploration of high dimensional data. Here our reference to high dimensions has a practical limit of 30 or so dimensions. The software has its roots in Explor4 (see Carr and Nicholson, 1988), but has evolved well beyond. ExplorN is authored by Qiang Luo, Edward J. Wegman, Daniel B. Carr and Ji Shen and is written in C and exploits the GL graphics library available on Silicon Graphics workstations. ExplorN combines the early work of Carr and Nicholson on stereo ray glyph plots with more recent multidimensional visualization tools such as parallel coordinates, Wegman (1990), d-dimensional grand tour, Wegman (1991) and saturation brushing, Wegman and Luo (1996). Multidimensional display is available in either a scatter plot matrix, a parallel coordinate plot, or a stereo ray glyph plot. Brushing is available in any of these displays and the brushed color becomes an attribute of the brushed data so that brushed color is linked to all other plots.

ExplorN uses a general d-dimensional grand tour rather than simply a two-dimensional grand tour. The results of the tour are available in all three forms of multidimensional display. The high interaction graphics allows one to temporarily suspend the tour, brush with color and then resume the grand tour. Two additional features are of interest. The color saturation may be varied. The idea is that the display may be brushed with very low color saturation levels, i.e. nearly black. The program uses the alpha-channel feature of Silicon Graphics workstations to add saturation levels. Thus in regions of heavy overplotting, color saturation is high, while in regions of little overplotting, color saturation is low. This feature is useful for very large data sets since the net effect is to produce a color-coded density plot. This density has the interesting feature that it does not require any smoothing (convolutions) and hence preserves edges and boundaries well. Coupled with a partial grand tour it can be used to produce tree structured decision rules. See Wegman and Luo (1996). The software runs on any SG workstation supporting and alpha-channel and 24 bit color. We typically run on a Silicon Graphics Onyx with RE^2 graphics engine. We have use ExplorN for data sets as large as 250,000 observations in 10 dimensions and so is capable of handling fairly massive data sets. ExplorN also produces three-dimensional rendered density surfaces using lighting models. These are described in more detail in Wegman and Carr (1993) and Wegman and Luo (1995). Some of this work is described at URL **http://www.galaxy.gmu.edu/papers/inter96.html** and some images of the rendered densities are available at URL **http://www.galaxy.gmu.edu/images/gallery/research_arcade.html**.

MANET

MANET is software for interactive statistical graphics running on a Macintosh computer. MANET is designed by Professor Antony Unwin, Chair of the Computer-oriented Statistics and Data Analysis Group, Institute for Mathematics at the University of Augsburg, Augsburg, Germany. The basic structure of the program was written by George Hawkins in 1994. Heike Hoffman and Bernd Siegel carried out the remaining programming since then implementing novel interactive features and adding innovative displays. Dr. Martin Theus has also contributed extensively to the design of the program.. MANET is written in C++ and provides standard interactive graphical features. It is similar in design to DataDesk or JMP. Unlike its commercial counterparts, however, the MANET software focuses on innovative methods for graphically dealing with missing values. To our knowledge, it is the only software that consistently attempts to represent missing values graphically. All graphics are fully linked and may be interacted with directly. MANET follows Macintosh conventions and is consistent with other Macintosh packages. It is an exploratory tool and is intended to be used with other more traditional software. Unlike ExplorN, MANET does not support massive data sets.

The current version is Version 0.1832. The next version, we are told by Professor Unwin, will be Version 0.1848. The version numbers correspond to important dates in the life of the impressionist painter, Manet, and presumably have the advantage that there cannot be an unlimited number of versions. All of the software developed by Professor Unwin's group at Augsburg is named for impressionist painters because the software is intended to give a visual impression of the data. MANET is freeware and may be obtained by sending email to **unwin@uni-augsburg.de**. More information on MANET is available at **http://www1.Math.Uni-Augsburg.DE/~theus/Manet/ManetEx.html**.

4. Future: Massive Datasets and SIFTWARE

The software described in Sections 2 and 3 in many ways reflects traditional thinking about data sets and data analysis. By this we mean that essentially they reflect mental world views of a fairly conventional nature about the size and dimensionality of data sets. Wegman (1995) articulated some issues of computational complexity in conjunction with data set sizes and discussed the limits of computational feasibility and well as visualization feasibility. This was motivated in part by considerations of NASA's EOS-DIS project as well as by implications of massive data sets available as a result of accumulations in financial transaction databases. Following Huber (1994), Wegman discusses a range of data set sizes ranging from tiny (10^2 bytes) to huge (10^{10} bytes) and even beyond this to the multi-terabyte data sets promised by EOS. It is clear that data sets of this magnitude test the computational limits as well as the visualization limits of all the software discussed in Sections 2 and 3. Automated or semi-automated accumulation of data has been a hallmark of space and astronomical experimental science for decades. In this section, we attempt to articulate the impact of computational and electronic instrumentation advances on what we conventionally think of as data analysis.

Except for those who are completely out of touch with current computer technology, the World Wide Web has become a nearly ubiquitous fact of daily life. It provides an essentially new learning paradigm, a virtual library at the finger tips of anyone with a network connected computer. Much like Marshall McCluhan's famous adage of the 1960's, the *Medium is the Message*, the web is the message and McCluhan's global village is in reality a global cyber-village. One can easily anticipate in much the same way that text material is available and searchable by the dozen or so indexed web search engines, that in the future databases of numerical and symbolic data will be searchable and retrievable through similar mechanisms. Just as there is a virtual text library today, there will be a virtual data library in the future. Some virtual data libraries in primitive forms are available even today; some cancer and related medical databases are privately held and, to a limited extent, fragmented pieces of genome databases are publicly available on the web. The widespread availability of easily searchable and retrievable databases would have the effect of accelerating research both for subject matter scientists who could rapidly verify or discard conjectures based on empirical evidence as well as for

methodological scientists such as statisticians who could test and refine methodologies based on application of their methods to real data.

We believe that this is a direction that statistical and computational research will take. Data acquisition, sorting and refinement will become part of the data analysis process. The phrase, *siftware*, we have coined in the title has its origins in a typographical error (*o* is next to *i* on the qwerty keyboard), but in fact massive databases (terabytes and larger) will not simply be one massive data set, but many, many somewhat smaller data sets. A terabyte database could easily be a million 10^6 data sets. However you slice it, this is not something that is feasible for an individual to browse in an afternoon. Thus, data analysis software must also be data siftware ... software designed to aid in isolating interesting worthwhile data sets for the researcher to examine. In this spirit, we discuss three tools we think will reflect future directions.

JAVA

JAVA is a programming language which represents an extension of the world wide web capabilities. Basic documents on the web are constructed using HTML, the hypertext markup language. HTML is a simple addition to ASCII text which allows the inclusion of simple formatting commands which are interpreted by the client browser. Most web pages are static in the sense that once a server delivers the HTML text to the browser, the server has done its job and the static text (and images and multimedia content) is interpreted and displayed by the client's browser. Most web pages are static in the sense that the web page is static in the server and also in the display by the client. There is a possibility of interaction between client and server through so-called CGI scripts. CGI is an abbreviation for common gateway interface. CGI scripts allow for the client to send information back to the server, for the server to carry out some action based on data from the client, and for the server to generate a web page dynamically based on the client's data. Typically the two most used application of CGI scripts are search requests for searching some database resident on the server or registration/purchase requests used, for example, to register for a scientific meeting. With a CGI script, the text is generated dynamically (on demand) by the server, but is still typically a static display in the client browser.

JAVA is a fully distributed, object oriented programming language which allows for the creation a fully interactive web-based system. Data and the tools to view it can be sent to the client browser. In object oriented programming, one creates objects instead of variables. Objects have attributes (values) and methods (subroutines). JAVA allows attributes and methods to be linked together. In particular, JAVA allows applets, small applications or subroutines, to be created and transmitted across the web just as static HTML documents are now transmitted. The applets run on the client machine rather than the server. A typical application might be to have a data set and a linked

statistical routine, for example a plotting routine or, say, a time series analysis routine. The routine would typically be interactive, so, for example, the graphic might be rotated dynamically or there might be a slider bar for the dynamic adjustment of a parameter in a time series model.

JAVA is similar to C++ is comparatively easy to learn for those familiar with C++. JAVA is distributed in the sense that JAVA applets can run on machines connected by networks. Communication between applets is possible and so in this sense there is a resemblance to a massively parallel computer. JAVA is both interpreted and compiled. The original source code is compiled to so-called *byte code*. Byte code is a machine neutral code which is interpreted locally by each client in a *JAVA Virtual Machine* (JVM). The JVM runs inside of a client browser and is isolated from the other resources of the client machine. JAVA byte code is interpreted by the client as it is loaded. JAVA is intended to be a secure system in that running inside the JVM although security problems do exist with present implementations. However, access to local data is restricted and the JAVA is a securable environment. The JAVA environment is architecturally neutral in that it will run on any machine that has a browser supporting a JVM. (JAVA is currently support in Netscape 2.01 browsers for machines with Unix and Windows95 operating systems.) Interpreted byte code is very fast, producing near machine speeds and typically byte code sizes are very small even for comparatively complex programs.

So why have we declared that JAVA is related to statistical/data analysis software of the future. We declare this as software to be watched because it is a practical implementation of a new paradigm in distributed computing, just as web browsers were a new paradigm in anonymous access to non-local machines. JAVA will allow for not only the distribution of text and multimedia, but also of computer applications and data. We could imagine an applet launched to search a distant database for a particular class of data (more on this in the section entitled METANET below) and to return to the client small subsets of the database which fulfill the search terms. The search applet could run in the background and could use heuristic search algorithms (sometimes called by statisticians *cognostics*), sort of a data analyst's analog to the military's fire and forget weapons. Moreover, under a JAVA framework, new statistical, data analytic and other methodologies could be made available over the web, and could be tried out by practitioner's in other research fields on their own data and their own computer. The possibilities are quite extensive and we have not yet seen even the most elementary of these uses implemented. More information on JAVA can be found at **http://java.sun.com/**. A AltaVista search on JAVA returns more than 10,000 websites referencing JAVA.

JAVA is a response to the enormous popularity of the world wide web. If we consider the possibility of extending the web in a natural way to acquiring data in the same way we acquire human-consumable information, new mechanisms must be sought to provide for the distribution of that data. The next two items, VDADC and METANET, are concepts that have been floated as methods for accessing and distributing data. The ideas amount to a substantive method for evolving our notion of siftware. These ideas are under development, but of course are not available for usage yet. They depend not only on technology developments, but also on political developments. The latter are much more problematic. We believe something akin to these will ultimately be developed.

VDADC

NASA has created a wonderfully vulnerable concept to deal with the massive data sets anticipated from the Earth Observing System (EOS). Called the DAAC, Distributed Active Archive Centers, NASA manages to encode two oxymorons in a single name (distributed centers and active archives). The DAACs are intended as central repositories for the massive amounts of data expected from EOS and, as such, form a prototype for other application fields with massive data sets. One proposal currently under development for access data in the DAACs is the Virtual Domain Application Data Center (VDADC). A VDADC is a way of organizing massive data sets to provide an optimal search for any particular group of users. The VDADC is designed to accommodate large data sets in the sense that a large number of descriptors (metadata) characterize the data, as opposed to a large quantity of data with relatively few characterizing parameters. The VDADC views its world of data as organized into distinct trees, without any specific linkages among the trees. Each tree is called a "database". The database is the most general parameter to describe the data. The early part of the search mechanism is to determine which database is the closest fit to the user request. The trees are constructed so that most searches are resolved in the least amount of time. The trees are dynamically constructed (or reconstructed) based on user queries. Thus, the trees provide the routes for the searches to occur, and these routes will evolve into the most optimal design as user queries are evaluated by the system. The trees are conceptually viewed as three level structures. The top level (the database level) is a gross organizing scheme; it also serves to define any number of parameters to be inherited by the levels below it. The middle level consists on any number of nodes which will form the search mechanism. The bottom layer consists descriptors pointing to the actual data.

Each of these distinct search trees will resolve queries for any of the data located in the bottom of that tree. The details of the data retrieval mechanism will be located within that search tree. Each node in the tree serves as part of the search mechanism and may also contain actual data or code for searching or eventual transmission to the user. Traversing that node will activate its code sections for that node, as well as its children. Therefore, this program will

allow users to make queries against disparate types of data bases. A plausible scenario might be a database of environmental factors, a database of normal cancer incidence, a database of patient files, and a final database of generic medical imagery with control images. A user may use wish to use the environmental data base to select a geographic region (matching a set of cancer causing parameters), and then return data from patient files that reside within that region. Since these are two very different types of data, the actual mechanism of the data return will vary, but that can be defined as part of each database search tree.

The data that is returned may need to be reformatted to fit a particular query. These reformatting routines may be included as part of the search tree (in the case of a database that has very non-standard data) or can be assumed will be done as part of the transport mechanism to the user. This would apply to standard type of image data. The search tree will only have to generate a tag describing the type of data the transport mechanism should return to the user. The result of a query might be a count of the data which would satisfy that query, a list of appropriate data sets, or a data stream. The transport of the data will be handled by a separate module, and the search tree will simply pass the code to effect the data conversion to that module, and the transport mechanism only has to evaluate that code.

It is envisioned that user queries will be generated from web pages. These web pages (forming the user queries) will be generated by the VDADC system itself to reflect the current state of the dynamic search trees. Since the forms are being generated by the system, this would allow a specific user request to be partially formulated by the system. If a user were to request a very narrow field of the data base, the query presented to the search mechanism will have actually been generated by the web page, and that query could include a number of hints to the search mechanism to allow a very fast descent through the search trees. The user query will then be broken into a number of independent queries and the appropriate result of those queries will be delivered to the user.

Each node of the tree will either be the definition of a data set, or a part of the search mechanism to find the correct data set. Since a user query might not penetrate to the very bottom of a search tree, all data sets located below the satisfaction of the user query will be returned as the answer for that query. For example, a data base may consist of a number of daily data sets. These data sets could then be grouped by weeks, which could be grouped by months, which could be grouped by quarters, which could be grouped by years. While the actual data is at a daily level, a request for data from any particular week would return the children of that week, which would be the correct daily sets. Each node pointing to one of the daily sets of data could also contain data itself, such as the average value for that data set. In the given example, each

level could write its own average return which returned the average of its children's average value tags, or the higher levels could assert code which forced the evaluation of an average from the actual data sets, instead of allowing the averages to propagate up the tree.

While the VDADC concept is specifically focused on NASA's EOS data, the METANET concept described below envisions a national and international digital data library which would be available via the Internet. We consider a heterogeneous collection of scientific databases.

METANET

Automated Generation of Metadata

In general, it is assumed there are metadata that describe file and variable type and organization, but that have minimal information on scientific content of the data. In the raw form, a data set and its metadata has minimal usability. For example, a satellite-based remote sensing platform will produce thousands of image data sets in the same file form based on the same instruments over the same geographic regions. However, only the image data sets with certain patterns in the image will be of interest to the scientist. Without additional metadata about the content, the scientist would have to scan all of these images, a daunting prospect for terabyte data sets. Thus a strategy for making the data usable is to link the data set to digital objects that are used to index the data set. The search operation for a particular structure in a data set then becomes a simple indexing operation on the digital objects linked to the data set. The idea is to link digital objects with scientific meaning to the data set at hand. The digital objects become part of the searchable metadata associated with the data set. It should be said that the goal of creating digital objects reflecting the scientific content of the data is not to replace the judgment of the scientist, but to narrow the scope of the data sets that the scientist must consider. It is quite possible that some of the patterns found by the automated methods will be inappropriate.

The key element is to automate the process of creating digital objects with scientific meaning to be linked to the data set. The digital objects will essentially be named patterns we find in the data sets. The concept is to have a background process, launched either by the database owner or, more likely, via applet created by the virtual data center (e.g. a VDADC), examining databases available on the dataweb and searching within data sets for recognizable patterns. When a pattern is found in a particular data set, the digital object corresponding to that pattern is made part of the metadata associated with that data set. Also pointers would be added to that metadata pointing to metadata associated with other distributed databases containing the same pattern. This metadata will be located in the virtual data center and through this metadata,

distributed databases will be linked. This linking is to be done on the fly as data is accumulated in the database. On existing databases, the background process would run as compute cycles are available. The idea is that because the database is dynamic, the background process would always be running adding metadata dynamically.

Patterns to be searched for are to be generated by one of at least three different methods, that is 1) empirical or statistical patterns, 2) model-based patterns, and 3) patterns found by clustering algorithms. By empirical or statistical patterns in the data, we mean patterns that have been observed over a long period of time that may be thought to have some underlying statistical structure. This could be a pattern which might be speculative and for which the scientist would like to have additional verification. Certain weather patterns such as hurricanes in late summer in the subtropical zones or certain protein patterns in DNA sequencing are examples of empirical or statistical patterns. Model-based patterns clearly are predictive and would be of interest if verified in real data. Statistical, empirical, and model-based patterns all originate with the scientists and have some intellectual imperative behind them. The patterns found by clustering methods by contrast are patterns which are found by purely automated techniques which may or may not have scientific significance. The idea is to flag for the scientist unusual patterns that bear further investigation. Statistical clustering methods have received considerable attention and extremely effective recursive, nonparametric methods might be employed to accomplish this task.

Query and Search

The idea of the automated creation of metadata is to develop metadata that reflects the scientific content of the data sets within the database rather just data structure information. The locus of the metadata is the virtual data center. The end user would see only the virtual data center. The original metadata, resident in the actual data centers, would be reproduced in the virtual center. However, that original metadata would be augmented by metadata collected by the automated creation procedures mentioned above, by pointers used to link related data sets in distributed databases, and by metadata collected in the process of interacting with system users.

The general desirata for the scientist is to have a comparatively vague question which can be sharpened as the scientist interacts with the system. For example, "give me data about pollution in the Chesapeake Bay" might be an initial query which would possibly be sharpened to something like "give me data about nitrate and ammonia concentrations in the Chesapeake Bay within 8 miles of the entry of waters from the Potomac River into the Bay." Clearly, even the second query is comparatively vague. It may be that data is accessible from several distributed databases for this type of query through the following

logic. Nitrates and ammonia support algae growth that responds to infrared. Therefore, an image data set in visible light available on one database may be compared to an image data set taken in infrared by a different instrument and available on a second database may be compared in order to show a high infrared-to-visible light intensity. This is indicative of robust algae growth and that indicates excess nitrate and ammonia concentrations. This excess ratio would be a statistical or possibly model-driven pattern which was already established by the automated generation of metadata mechanism discussed earlier. Thus the retrieval process consists of not only a browser mechanism for requesting data when the user has a precise query, but should also support an expert system query capability which will help the scientist reformulate a vague question in a form that may be submitted more precisely.

Query and search would contain four major elements: 1) client browser, 2) expert system for query refinement, 3) search engine and 4) reporting mechanism. The first and last are relatively straightforward.

Client Browser The client browser would be a piece of software running on the scientist's client machine. The client machine is likely to be a PC or a workstation. This component is straightforward. The idea is to have a GUI interface that would allow the user to interact with a more powerful server in the virtual data center. The client software is essentially analogous to the myriad of browsers available for the world-wide web.

Expert System for Query Refinement There are two basic scenarios for the interaction of the scientist with the server: first, the scientist knows precisely the location and type of data he desires, and second, he knows generally the type of question he liked to ask, but has little information about the nature of the databases with which he hopes to interact. The first scenario is comparatively straightforward, but the expert system would still be employed to keep a record of the nature of the query. The idea is to use the queries as a tool in the refinement of the search process. The second scenario, however, is the more complex. The approach is to match a vague query formulated by the scientist to one or more of the digital objects discovered in the automated-generation-of-metadata phase. The expert system would initially be given rules devised by discipline experts for performing this match. Given an inquiry, the expert system would attempt to match the query to one or more digital objects (patterns). It would provide the scientist with an opportunity to confirm the match or to refine the query. This interplay would continue until the scientist is satisfied with the proposed matches. The expert system would then engage the search engine in order to synthesize the appropriate data sets. The expert system would also take advantage of the interaction with the scientist to form a new rule for matching the original query to the digital objects developed in the refinement process.

Thus there are two aspects: one is the refinement of the precision of an individual search and the other is the refinement of the search process. Both aspects have the same goal; one is tactical and the other is strategic. The refinement would be greatly aided by the active involvement of the scientist. The scientist would be informed how his particular query was resolved; this allows him to reformulate the query efficiently. The log files of these iterative queries would be processed automatically to inspect the query trees and possibly, improve their structure.

Also there are two other considerations of interest. First, other experts not necessarily associated with the data repository itself may have examined certain data sets and have commentary in either informal annotations or in the refereed scientific literature. These commentaries should form part of the metadata associated with the data set. Part of the expert system should provide an annotation mechanism that allows users to attach commentary or library references (particularly digital library references) as metadata. Obviously, such annotations may be self-serving and potentially unreliable. However, the idea is to alert the scientist to information that may be of use. User derived metadata would be considered secondary metadata.

The other consideration is to provide a mechanism for indicating reliability of data. This would be attached to a data set as metadata, but may in fact be derived from the original metadata. For example, a particular data collection instrument may be known to have a high variability. Thus any data set which is collected by this instrument, no matter where in the database it occurs, should have as part of the attached metadata an appropriate caveat. Thus the concept of automated collection of metadata should have a capability to not only examine the basic data for patterns, but also examine the metadata itself and based on collateral information such as just mentioned, be able to generate additional metadata.

Search Engine As indicated above, large scale scientific information systems will likely be distributed in nature and contain not only the basic data but both structured metadata, for example, sensor type, sensor number, measurement date and unstructured metadata, for example, a text-based description of the data. These systems will typically have multiple main repository sites that together will house a major portion of the data as well as some smaller sites, virtual data centers, containing the remainder of the data. Clearly, given the volume of the data, particularly within the main servers, high performance engines that integrate the processing of the structured and unstructured data would be required to support desired response rates for user requests.

Both DBMS and information retrieval systems provide some functionality to maintain data. DBMS allow users to store unstructured data as binary large

objects (BLOB) and information retrieval systems allow users to enter structured data in zoned fields. However, DBMS offer only a limited query language for values that occur in BLOB attributes. Similarly, information retrieval systems lack robust functionality for zoned fields. Additionally, information retrieval systems traditionally lack efficient parallel algorithms. Using a relational database approach to information retrieval allows for parallel processing since almost all commercially available parallel engines support some relational database management system. An inverted index may be modeled as a relation. This treats information retrieval as an application of a DBMS. Using this approach, it is possible to implement a variety of information retrieval functionality and achieve good run-time performance. Users can issue complex queries including both structured data and text.

The key hypothesis is that the use of a relational DBMS to model an inverted index will: 1) Allow users to query both structured data and text via standard SQL. In this fashion, users may use any relational DBMS that supports standard SQL; 2) Allow implementation of traditional information retrieval functionality such as Boolean retrieval, proximity searches, and relevance ranking, as well as non-traditional approaches based on data fusion and machine learning techniques; 3) Take advantage of current parallel DBMS implementations so that acceptable run-time performance can be obtained by increasing the number of processors applied to the problem.

Reporting Mechanism The basic idea is not only to retrieve data sets appropriate to the needs of the scientist, but also to scale down the potentially large databases the scientist must consider. That is, the scientist would consider megabytes instead of terabytes of data. The search and retrieval process may still result in a massive amount of data. The reporting mechanism would thus initially report the nature and magnitude of the data sets to be retrieved. If the scientist agrees that the scale is appropriate to his needs, the data will be delivered by an FTP or similar mechanism to his local client machine or to another server where he wants the synthesized data to be stored.

5. General Comments

In this paper, we have tried to provide general assessments and pointers to a variety of statistical, data analysis and related software that would appear to address some of the needs of astronomers and space scientists. This is, of course, a highly personalized view of the statistical software world. We attempt to represent a variety of opinions by drawing in a rather larger number of authors than might be typical for a statistics paper. We have divided our discussion into commercial, academic research and as-yet-not-developed software. The intent is to provide a broader vision of software, not to merely catalog a dozen or so packages.

The use of pointers (URLs) from the world wide web extends the utility of this paper by giving references that are likely to be dynamically updated. Particularly with commercial software, the vendors have a significant stake in maintaining current information while at the same time ensuring the continuity of the web page address. We believe that these URLs may be the most valuable aspect of this summary paper. Also of note is the number of hits a particular software gets under an AltaVista search. This search engine is by far the most comprehensive of all the search engines available. We believe a reasonable inference is that a particular software's popularity (utility?)(market penetration?) is proportional to the number of web pages devoted to it. This gives the reader a gauge of the success of a piece of software.

Finally we want to note some addition URLs that may be of general interest for both the astronomy and the statistics audiences. Most statisticians know of STATLIB available at URL **http://lib.stat.cmu.edu/**. STATLIB is an extraordinarily comprehensive resource for data, software, and other information for the statistics community. It is a highly recommended site to visit and browse. A visit to the URL at Cornell University, **http://www.stat.cornell.edu/compsites.html**, yields a series of pointers to a variety of software and computing resources. A guide to statistical computing resources on the net can be found at **http://asa.ugl.lib.umich.edu/chdocs/statistics/ stat_guide_home.html**.

References

Asimov, D. (1985) "The grand tour: A tool for viewing multidimensional data," ***SIAM Journal on Scientific and Statistical Computing***, 6(1), 128-143.

Babu, G. J. and Feigelson, E. D. (1996) "Spatial point processes in astronomy," ***Journal of Statistical Planning and Inference***, 50(3), 311-326.

Becker, R. A., Chambers, J. M. and Wilks, A. R. (1988) ***The New S Language***, Wadsworth and Brooks/Cole, Pacific Grove, CA.

Becker, R. A. and Cleveland, W. S. (1984) "Brushing a scatterplot matrix: high interaction graphical methods for analyzing multidimensional data," Technical Memorandum, AT&T Bell Laboratories, Murray Hill, NJ.

Buja, A, Cook, D. and Swayne, D. F. (1996) "Interactive high-dimensional data visualization," ***Journal of Computational and Graphical Statistics***, 5(1), 78-99.

Carr, D. and Nicholson, W. (1988) "Explor4: A program for exploring four-dimensional data using stereo-ray glyphs, dimensional constraints, rotation and masking," In ***Dynamic Graphics for Statistics***, (W. S. Cleveland and M. E. McGill, eds.) Wadsworth Inc., Belmont CA., 309-329.

Chambers, J. M. and Hastie, T. K., (eds.) (1992) ***Statistical Models in S***, Wadsworth and Brooks/Cole, Pacific Grove, CA.

Cleveland, W. S. and McGill, M. E. (1988) ***Dynamic Graphics for Statistics***, Wadsworth Inc., Belmont CA.

Donoho, D., Huber, P. J. and Thoma, H. (1981) "The use of kinematic displays to represent high dimensional data," ***Computer Science and Statistics: Proceedings of the 13th Symposium on the Interface***, 274-278, Springer-Verlag, New York.

Feigelson, E. D. and Babu J. G. (eds.) (1993) ***Statistical Challenges in Modern Astronomy***, Springer-Verlag, New York.

Fishkeller, M. A., Friedman, J. H. and Tukey, J. W. (1974) "PRIM-9: An interactive multidimensional data display and analysis system," SLAC-PUB-1408, Stanford Linear Accelerator Center, Stanford, CA

Friedman, J. H. and Tukey, J. W. (1974) "A projection pursuit algorithm for exploratory data analysis," ***IEEE Transactions on Computers***, C-23, 881-890.

Francis, Ivor (1981) ***Statistical Software: A Comparative Review***, North Holland: New York.

Furnas, G. W. and Buja, A. (1994) "Prosection views, dimensional inference through sections and projections" (with discussion), ***Journal of Computational and Graphical Statistics***, 3, 323-385.

Hayes, Annie (1982) ***Statistical Software: A Survey and Critique of its Development***, Office of Naval Research, Arlington, VA

Huber, Peter J. (1994) "Huge data sets," ***COMPSTAT: Proc. in Computat. Statist.***, 11th Symp., 3-13, (Dutter, R.; Grossmann, W. eds.) Physica-Verlag, Heidelberg.

Jaschek, C. and Murtagh, F. (eds.) (1990) ***Errors, Bias and Uncertainties in Astronomy***, Cambridge University Press, Cambridge.

Majure, J. J., Cook, D., Cressie, N., Kaiser, M., Lahiri, S. and Symanzik, J. (1995) "Spatial CDF estimation and visualization with application to forest health monitoring," ***Computing Science and Statistics***, 27, (Rosenberger, J. and Meyer, M., editors).

McDonald, J. A. (1982) ***Interactive graphics for data analysis***, Ph.D. dissertation, Stanford University, Stanford, CA.

Murtagh, F. and Heck, A. (eds.) (1988) ***Astronomy from Large Databases - Scientific Objectives and Methodological Approaches***, ESO Conference and Workshop Proceedings, No. 28.

Rolfe, E. J. (ed.) (1983) ***Statistical Methods in Astronomy***, European Space Agency Special Publication ESA SP-201.

Statistical Sciences (1993) ***Statistical Analysis in S-Plus***, Version 3.1, Seattle: StatSci, a division of MathSoft, Inc.

Swayne, D.F., Cook D. and Buja, A. (1992). ***User's Manual for XGobi, A Dynamic Graphics Package for Data Analysis***, Bellcore Technical Memorandum.

Symanzik, J., Majure, J. J. and Cook, D. (1995) "Dynamic graphics in a GIS: A bi-directional link between ArcVIew 2.0 and XGobi, ***Computing Science and Statistics***, 27, (Rosenberger, J. and Meyer, M., editors).

Tierney, Luke (1990) ***Lisp-Stat***, John Wiley and Sons, New York.

Trumpler, R. J. and Weaver, H. F. (1953) ***Statistical Astronomy***, University of California Press, republished in 1962 by Dover, New York

Venables, W. N. and Ripley, B. D. (1994) ***Modern Applied Statistics with S-PLUS***, Springer-Verlag, New York.

Wegman, E. J. (1990) "Hyperdimensional data analysis using parallel coordinates," ***J. American Statist. Assoc.***, 85, 664-675.

Wegman, E. J. (1991) "The grand tour in k-dimensions," ***Computing Science and Statistics: Proceedings of the 22nd Symposium on the Interface,*** 127-136.

Wegman, E. J. (1995) "Huge data sets and the frontiers of computational feasibility," ***Journal of Computational and Graphical Statistics***, 4(4), 281-195.

Wegman, E. J. and Carr, D. B. (1993) "Statistical graphics and visualization," in ***Handbook of Statistics 9: Computational Statistics,*** (Rao, C. R., ed.), 857-958, North Holland, Amsterdam.

Wegman, E. J. and Hayes, A. R. (1988) "Statistical software," ***Encyclopedia of Statistical Sciences***, 8, 667-674.

Wegman, E. J. and Luo, Qiang (1995) "Visualizing densities," Technical Report 100, Center for Computational Statistics, George Mason University, Fairfax, VA

Wegman, E. J. and Luo, Qiang (1996) "High dimensional clustering using parallel coordinates and the grand tour," Technical Report 124, Center for Computational Statistics, George Mason University, Fairfax, VA

SF 298 MASTER COPY KEEP THIS COPY FOR REPRODUCTION PURPOSES

REPORT DOCUMENTATION PAGE

Form Approved
OMB NO. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
May, 1996

3. REPORT TYPE AND DATES COVERED
Technical

4. TITLE AND SUBTITLE
Statistical Software, Siftware and Astronomy

5. FUNDING NUMBERS
DAAH04-94-G-0267

6. AUTHOR(S)
Edward J. Wegman, Daniel B. Carr, R. Duane King, John J. MIller, Wendy L. Poston, Jeffrey L. Solka and John Wallin

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(ES)
Center for Computational Statistics
George Mason University
Fairfax, VA 22030

8. PERFORMING ORGANIZATION REPORT NUMBER
#128

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSORING / MONITORING AGENCY REPORT NUMBER
ARO 32850.12-MA

11. SUPPLEMENTARY NOTES
The views, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy or decision, unless so designated by other documentation.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12 b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*
This paper discusses staitstical, data analytic and related software that is useful in the realm of astronomy and spaces sciences. The paper does not seek to be comprehensive, but rather to present a cross section of software used by practicing statisticians. The general layout is first to discuss commercially available software, then academic research software and finally some possible future directions in the evolution of data-oriented software. We specifically exclude commercial database software from the discussion, although it is relevant. The paper focuses on providing internet (world wide web) pointers for a variety of the software discussed.

14. SUBJECT TERMS Spatial statistics, exploratory analysis, scientific databases, metanetworking

15. NUMBER IF PAGES
31

16. PRICE CODE

17. SECURITY CLASSIFICATION OR REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18
298-102

GENERAL INSTRUCTIONS FOR COMPLETING SF 298

The Report Documentation Page (RDP) is used in announcing and cataloging reports. It is important that this information be consistent with the rest of the report, particularly the cover and title page. instructions for filling in each block of the form follow. It is important to ***stay within the lines*** to meet ***optical scanning requirements.***

Block 1. Agency Use Only *(Leave blank)*

Block 2. Report Date. Full publication date including day, month, and year, if available (e.g. 1 Jan 88). Must cite at least year.

Block 3. Type of Report and Dates Covered. State whether report is interim, final, etc. If applicable, enter inclusive report dates (e.g. 10 Jun 87 - 30 Jun 88).

Block 4. Title and Subtitle. A title is taken from the part of the report that provides the most meaningful and complete information. When a report is prepared in more than one volume, repeat the primary title, add volume number, and include subtitle for the specific volume. On classified documents enter the title classification in parentheses.

Block 5. Funding Numbers. To include contract and grant numbers; may include program element number(s), project number(s), task number(s), and work unit number(s). Use the following labels:

C	- Contract	**PR**	- Project
G	- Grant	**TA**	- Task
PE	- Program Element	**WU**	- Work Unit Accession No.

Block 6. Author(s). Name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. If editor or compiler, this should follow the name(s).

Block 7. Performing Organization Name(s) and Address(es). Self-explanatory.

Block 8. Performing Organization Report Number. Enter the unique alphanumeric report number(s) assigned by the organization performing the report.

Block 9. Sponsoring/Monitoring Agency Name(s) and Address(es). Self-explanatory.

Block 10. Sponsoring/Monitoring Agency Report Number. *(If known)*

Block 11. Supplementary Notes. Enter information not included elsewhere such as; prepared in cooperation with...; Trans. of...; To be published in.... When a report is revised, include a statement whether the new report supersedes or supplements the older report.

Block 12a. Distribution/Availability Statement. Denotes public availability or limitations. Cite any availability to the public. Enter additional limitations or special markings in all capitals (e.g. NORFORN, REL, ITAR).

- **DOD** - See DoDD 4230.25, "Distribution Statements on Technical Documents."
- **DOE** - See authorities.
- **NASA** - See Handbook NHB 2200.2.
- **NTIS** - Leave blank.

Block 12b. Distribution Code.

- **DOD** - Leave blank
- **DOE** - Enter DOE distribution categories from the Standard Distribution for Unclassified Scientific and Technical Reports
- **NASA** - Leave blank.
- **NTIS** - Leave blank.

Block 13. Abstract. Include a brief *(Maximum 200 words)* factual summary of the most significant information contained in the report.

Block 14. Subject Terms. Keywords or phrases identifying major subjects in the report.

Block 15. Number of Pages. Enter the total number of pages.

Block 16. Price Code. Enter appropriate price code *(NTIS only).*

Block 17. - 19. Security Classifications. Self-explanatory. Enter U.S. Security Classification in accordance with U.S. Security Regulations (i.e., UNCLASSIFIED). If form contains classified information, stamp classification on the top and bottom of the page.

Block 20. Limitation of Abstract. This block must be completed to assign a limitation to the abstract. Enter either UL (unlimited) or SAR (same as report). An entry in this block is necessary if the abstract is to be limited. If blank, the abstract is assumed to be unlimited.